PROCEEDINGS

55. IWK

Internationales Wissenschaftliches Kolloquium
International Scientific Colloquium

13 - 17 September 2010

Crossing Borders within the ABC

Automation,

Biomedical Engineering and

Computer Science

Faculty of
Computer Science and Automation

www.tu-ilmenau.de

TECHNISCHE UNIVERSITÄT
ILMENAU

Home / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=16739

Impressum
Published by

Publisher:	Rector of the Ilmenau University of Technology Univ.-Prof. Dr. rer. nat. habil. Dr. h. c. Prof. h. c. Peter Scharff
Editor:	Marketing Department (Phone: +49 3677 69-2520) Andrea Schneider (conferences@tu-ilmenau.de) Faculty of Computer Science and Automation (Phone: +49 3677 69-2860) Univ.-Prof. Dr.-Ing. habil. Jens Haueisen
Editorial Deadline:	20. August 2010
Implementation:	Ilmenau University of Technology Felix Böckelmann Philipp Schmidt

USB-Flash-Version.

Publishing House:	Verlag ISLE, Betriebsstätte des ISLE e.V. Werner-von-Siemens-Str. 16 98693 Ilmenau
Production:	CDA Datenträger Albrechts GmbH, 98529 Suhl/Albrechts
Order trough:	Marketing Department (+49 3677 69-2520) Andrea Schneider (conferences@tu-ilmenau.de)

ISBN: 978-3-938843-53-6 (USB-Flash Version)

Online-Version:

Publisher:	Universitätsbibliothek Ilmenau ilmedia Postfach 10 05 65 98684 Ilmenau

Home / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=16739

USE OF THE MECHATRONICS DEVELOPMENT SYSTEM EASYKIT FOR DIDACTICAL PURPOSES

H. Bönicke[1] *and C. Ament*[2]

[1] Department of Systems Analysis, Ilmenau University of Technology, Ilmenau, Germany
(Tel : +49-3677-69-1508; E-mail: holger.boenicke@tu-ilmenau.de)
[2] Department of Systems Analysis, Ilmenau University of Technology, Ilmenau, Germany
(Tel : +49-3677-69-2815; E-mail: christoph.ament@tu-ilmenau.de)

ABSTRACT

In the area of the design of mechatronical systems it is difficult to find well educated employees in a sufficient number. One reason is that this subject is very complex, so it is taught in schools very late or not at all. EasyKit is a development system for mechatronical systems, which also can be used for educational purposes. The electronics of the EasyKit system, the software EasyLab and the methods of developing mechatronical systems are presented in this paper as well as the didactical thoughts behind the EasyKit system. It is also presented how schools can use this system to already teach children at the age of about 15 years. The system was donated to selected schools, where teachers were asked to give a feedback about the learned knowledge. Statistical data about the feedback is also presented in this paper.

Index Terms **-** didactics, embedded, systems, microcontroller, graphical, software-design, modular, electronics

1. INTRODUCTION

Today's industry needs to enhance its products to be competitive on the market. The companies need to provide a fast response according to the wishes of the customers and at the same time they need to reduce the costs for prototype design and limited-lot production. This especially applies to traditionally mechanical or electrical systems like pumps or motors, where nowadays electronics need to be added, creating a mechatronical subsystem, which can be adjusted in the required way, to be integrated into other, bigger machinery, such as production lines. Especially electromechanical parts like pumps need to be controlled today. In former times many pumps were always pumping at 100% of power. If a lower pressure or flow was required, simply a valve was closed, but the pump was still working at full power. Of course this was a waste of energy and could cause damage to the pumps, the valves and other parts of the system. With the integration of electronics, the situation changed. Today the speed can be controlled very precisely and reliably. Normally the speed is controlled depending on the values of integrated sensors, using simple controllers like PID. For this purpose usually stored program controls (SPC) were used, adding a kind of intelligence to the machinery and controlling it entirely. In the last years, this intelligence was decentralized by adding microcontrollers to the subsystems. So the stored program control just sends a desired value to the microcontrollers of the subsystems. The microcontroller sets this value automatically, depending on values of sensors, which are very often already integrated into the small systems. It is probable that in the future more and more customer adapted systems with decentralized intelligence will be needed. To offer such products at a reasonable price, the companies need a sufficient amount of well educated employees. Unfortunately, many companies, especially small ones, do not have this kind of adequate educated employees for this purpose. The main reason for that is that students and pupils just get in contact with electronics very late or if they get in contact with it, the educational effect is not very high. This is mainly because the first contact with electronics is very difficult, because besides the electronics hardware, also the software needs to be understood, which normally means learning a text based programming language like C or Assembler. Another problem is that often no mechanical system is available, which is interesting enough for young people, to keep their motivation.

At this point EasyKit can be used for two tasks. On the one hand, with EasyKit, mechatronical systems of small and medium complexity can be developed without expert knowledge in electronics and text based software design. So the developer can use this tool for saving time. On the other hand, EasyKit can be used to lower the barrier of getting in first contact with electronics. With its additional documentation children learn to program a microcontroller with a graphical interface and they also learn, why they have to program it in a certain way. With this knowledge it will be easier to learn more about electronics and software later. Besides, children learn that actually electronics is not such a difficult subject as expected.

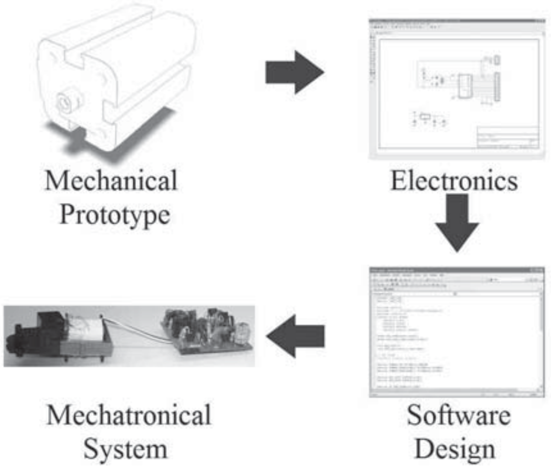

Figure 1 Traditional design method of mechatronical systems

2. EASYKIT AND EASYLAB

EasyKit and EasyLab are the basic tools, which can be used for getting in contact with microcontrollers for the first time. Originally, the system was developed to assist in microcontroller design, but because of its simple structured design, it is a good basis to create a didactical tool. But for this purpose, additional components needed to be added.

2.1. **EasyKit vs. traditional development methods**

The traditional way of designing mechatronical systems is a sequential workflow, which can be found in Figure 1 [1]. The sequential workflow increases the development time and also the expenses. In the first step, the mechanical prototype needs to be designed. For a good result, a mechanical engineer or another specialist is needed. At this point sensors and actuators have to be arranged on the mechanical system. For these sensors, in the second step, the electronics need to be designed. This electronics interacts with the sensors and the actuators of the system to add intelligence to the system, which can be used, e.g. to design a closed loop. For this step, a specialist in electronics or a similar subject is needed, if the result is supposed to be reliable. In the last step, the microcontroller of the electronics needs to be programmed. Of course for this purpose a specialist is required, too. Sometimes it is possible to find persons who can cover the electronics and the software part in a sufficient way, but very often for this traditional workflow at least three specialists are required to design a good product even if the final product has only a small complexity like a pump or a motor. These specialists need to be available, which is one problem, especially in smaller companies. The second problem is the sequential workflow. It is necessary, because during the second step the electronics can be tested together with the mechanical hardware and during the third step, the software can be tested on the whole system. This is needed to increase the reliability [2].

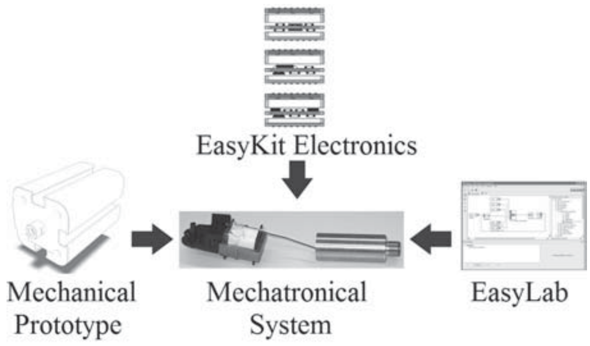

Figure 2 EasyKit development method

Anyways, especially the design of the electronics and the software is still error-prone and time consuming, compared to new methods, using modular hard- and software approaches.

2.2. **The EasyKit method**

By changing the development process, according to Figure 2, the described problems can be solved. Because of the modular electronics components and the according modular software components, used with a graphical interface, just a small amount of electronics and software knowledge is required to build a system of medium or small complexity. Besides the software designers can schedule their time more flexibly, because, after the interfaces are defined, the electronics and the software can be already prepared and tested.

2.3. **The electronics components of EasyKit**

To reach a sufficient flexibility, while using already designed electronics, the electronics components were modularized as shown in Figure 3. Each of the components fulfills its own task, is pin compatible to the others and was tested in various circuits. The EasyKit system electronics includes power supply components, CPU components and a set of interface components. The power supply components transform an external power supply for the microcontroller and contain interfaces for the communication to a PC or to other systems. The CPU components contain the microcontrollers and circuits to drive them as well as transceivers for communication purposes. The interface components have a variety of purposes.

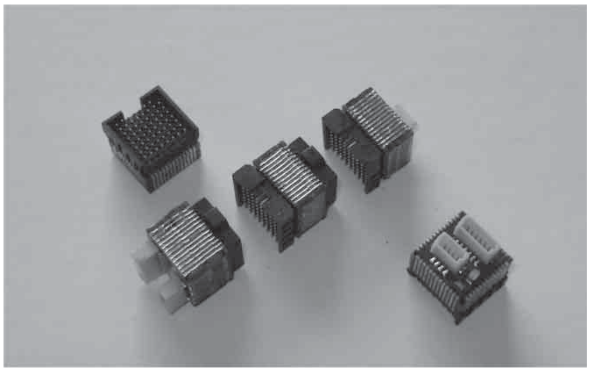

Figure 3 EasyKit electronics components

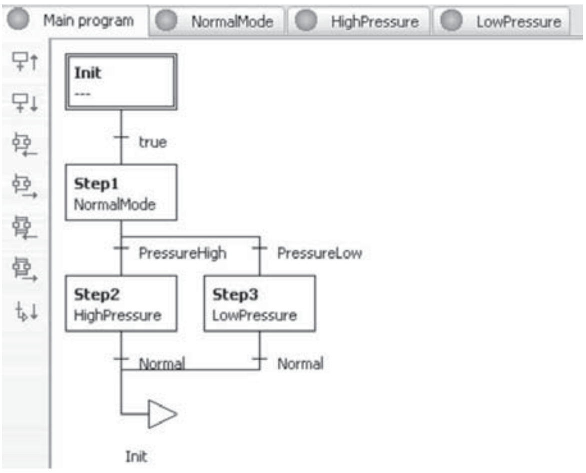

Figure 4 Structured Flow Chart (SFC), in this case used for a closed loop to control the pressure and flow of a pump system

Some of the components have high resolution analog digital changer, while others can be used for driving inductive loads. All of them contain integrated protection circuits. To connect the interface components to sensors and actuators easily, the components contain connectors.

Of course not every possible circuit could be preconceived. Especially the interface components are mostly designed for standard sensor and actuator interfaces, because the amount of interfaces, especially of digital serial signals, is almost unlimited. Mostly analog inputs and analog outputs are supported. Digital inputs and outputs are just supported as Boolean, not with a serial protocol. Because of these reasons, the mechanical and electrical layouts were disclosed [3], giving the user the possibility to design components himself. Of course when designing an own component, a specialist is required, to design this circuit. But the advantage would be that it is just this component which needs to be designed, not the whole circuit. In the future the component can be reused for other purposes, saving time later.

2.4. **The Software - EasyLab**

To design a program for the microcontroller and to burn it to it without extensive expert knowledge, the development tool EasyLab was created. The microcontroller program is designed in a graphical interface in two levels. On the upper level, a structured flow chart (SFC) is created, as shown in Figure 4 and on the lower level a synchronous data flow (SDF) is designed, as shown in Figure 5 [4]. The design of the SFC programs is mostly known from stored program controls. Various states are created and conditions are defined which determine the situation, when and how to change between the states. Each state is represented by an SDF program, which is combined of modular software components by drag and drop, which can be connected, similar to Simulink® [5] or LabView [6] [7].

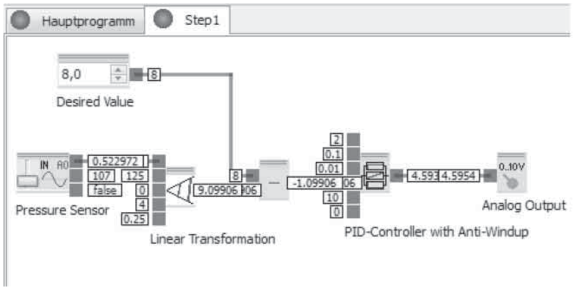

Figure 5 Example for a Synchronous Data Flow (SDF) program designed in EasyLab, in this case used for a closed loop of a speed control of a ventilation system

To design the SDFs, there exist many software components libraries. The basic libraries contain a wide variety of mathematical, logical, comparison and hardware related operations. Besides, there are also libraries for more complex tasks included, e.g. a library for automation purposes.

Similar to the electronics interfaces, the software components can also only include software for standard interfaces because of the wide variety of interfaces. Especially when thinking about digital interfaces with serial protocols, it becomes clear that not every protocol can be implemented. This results from the usage of graphical programming languages, which is essential for the simple usability. Anyways, it is possible to implement new protocols, because the software layout of the libraries was disclosed and an introduction of how to implement new components is included in documentation of EasyLab.

3. A DIDACTICAL STRATEGY TO GET TO KNOW ELECTRONICS

As already mentioned, EasyKit was developed primarily to be a development tool for industrial purposes, which could be used for developing mechatronical systems without extensive expert knowledge about electronics and software design. Anyways, another important thought of developing the EasyKit system was to create a didactical system, which could be used to teach the basics of designing electronics and software for mechatronical systems. While the EasyKit system and its software EasyLab are sufficient for the industrial purpose, they are not sufficient for the didactical purpose, but it is a good point to start at. For the didactical purpose, a variety of additional information was added to the EasyKit system and even a new test board for beginners was designed.

3.1. **Learning about microcontrollers**

Although the EasyKit system is really easy to use, the user has to know at least the basics of what a microcontroller can do and how it is working. This especially applies, if these basics should be taught to younger people or to other people, who never got in contact with this subject before.

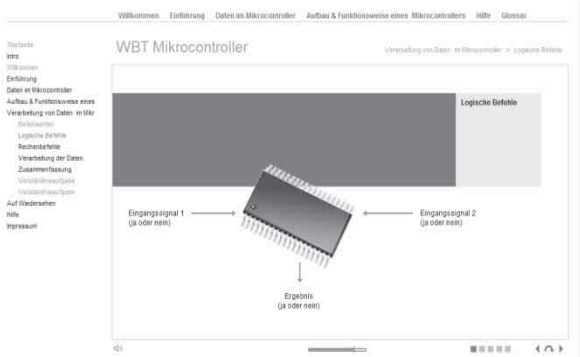

Figure 6 Screenshot of the Web Based Training used for teaching the basics of microcontrollers

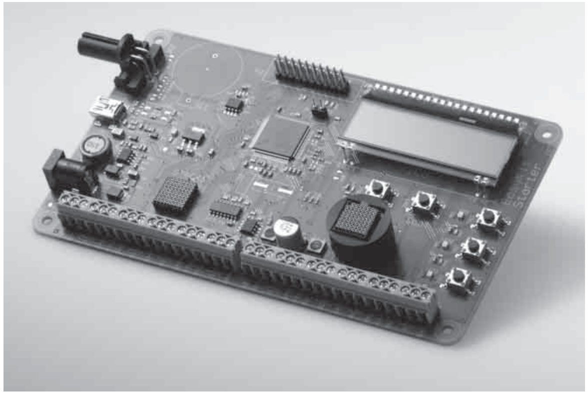

Figure 7 Starter board of the EasyKit system

To teach these people at least the basics of electronics and software, a web based training (WBT) was designed. The purpose was not to teach a certain programming language or to give the people the ability to design electronics on their own. It is just a first introduction to understand the processes inside of microcontrollers.

The WBT was designed for pupil around the age of 15 years. Another one, for professional education is already planned. To accomplish the lesson of the WBT, about one hour is scheduled. So in a typical lesson of 90 minutes with a break of five minutes, the WBT can be finished and the teacher has still enough time to give a small introduction and to assist slower pupil. The content of the WBT is explained by a spokesman and contains various examples during the chapter and a summary with small exercises in the end of every chapter, which causes it not to be boring even for younger children.

The WBT starts with information about where microcontrollers are used in our daily life and what tasks they take over, explained on the example of a coffee machine. This chapter is necessary for the user to understand that today basically in everything microcontrollers can be found. This should be a motivation to get in a closer contact with this subject.

The second chapter explains how data is stored inside the microcontroller and what data types are typically used. In this part it is important to understand, that the communication with the microcontroller has to be carried out in a different way, than the communication with another human. Typical numerating systems of microcontrollers are presented and it is shown that numbers need to be calculated from one system to another. Of course just superficial information is given, because the difficulty level needs to be adapted also to younger users.

The third chapter consists of information about the structure of microcontrollers. Most information is about the theoretical architecture of the core, of the memory and of the I/O-modules. The main focus is on the I/O-modules to understand how the microcontroller communicates with its environment.

The architecture is explained with the help of the already known example of the coffee machine. It is shown how the sensors are read, how the information is stored and processed in the memory and in the core, and how these values are used to set actuators. The information was cut to a useful limit, to make sure that the users still stay interested. Anyways, for people, who are interested in this topic, there exist some additional pages in this chapter. On these additional pages, more information is given about the architecture, e.g. how an analog digital changer is working.

The last chapter of the WBT is about logical and mathematical operations in the microcontroller. A variety of operations is presented and explained. The users of the WBT can calculate simple examples inside the WBT and can understand how the calculations take place.

The content of the WBT is not as extensive as a schoolbook, which was never the intention. It was just intended, to give the children a first impression of microcontrollers. For the next step, to use EasyKit and EasyLab, it is not necessary to remember everything from the WBT. It is just important to get a basic feeling about, how a microcontroller works.

3.2. A first test application with a simple program

After the WBT is accomplished successfully and the user has understood the basics about microcontrollers, it is the right time to try a first test application. For this purpose a starter board [8], as shown in Figure 7, was designed. It is based on the EasyKit infrastructure and can be programmed with EasyLab. The board cannot be as flexible as the final EasyKit system for the industrial purposes, e.g. the board uses a Cortex-M3 microcontroller of the STM32F family [9], which cannot be replaced. The communication and programming interface is realized through a USB connector, where the system also gets its power supply from. The interfaces of the microcontroller to sensors and actuators are already on the board, which increases the dimensions of the board, compared to the industrial EasyKit system.

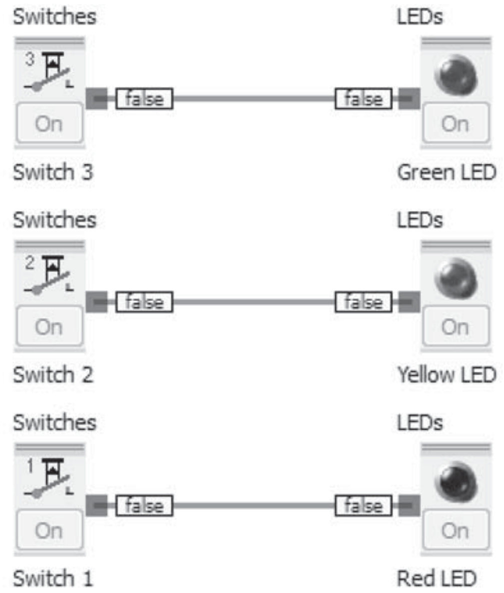

Figure 8 The first and most simple test program of the starter board of EasyKit

Besides these components, which are based on the industrial EasyKit system, the board contains switches and light emitting diodes of different colors, which can be used, to design small test programs with various complexities, e.g. a traffic light simulation. This option is implemented, because the user should have a very quickly feeling of success, when designing the first test program with the board and EasyLab, giving the user motivation to continue.

To put the board into operation, EasyLab needs to be installed and the board needs to be plugged to the USB of the PC. An installation manual and a quick start guide are included, which also give short information about handling the board. In the documentation, there are also small test exercises with detailed instructions to solve them, which are intended to get used to the handling of EasyKit. To test the programs, the integrated simulation tool can be used, before the program is loaded to the board. Like this, not every computer needs a connected starter board, when designing the program. So, even homework exercises are possible, because the software can be downloaded freely from the internet. The software for the controller can be designed and tested at home and the result can be compared at school. After the exercises are solved or if they are too difficult for the user, the solutions can be found in the program folder of EasyLab. One of these programs can be found in Figure 8. It simply reads the values of the switches and sets the light emitting diodes according to the switches. To put this program into operation and to get the first results from this "Hello World" program, just a few minutes are required. Compared to other development tools, the user gets the feeling of success very fast and stays motivated.

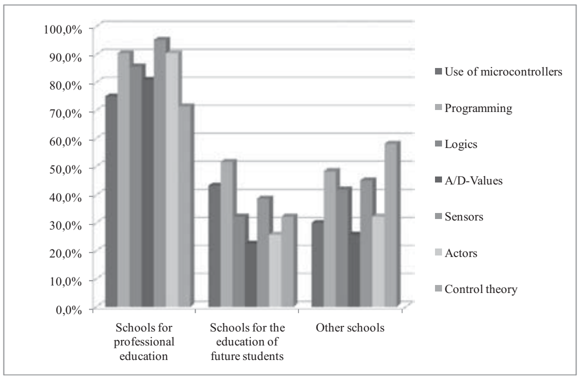

Figure 9 Statistical data of the questionaire given to the teachers, about the topics of the schools education

3.3. **Teaching the teachers**

The starter board was designed to be used in schools. For a first validation of the concept and the board, about 900 boards were given to almost 300 German schools. Additionally, a workshop was offered to the teachers. First they got a small introduction and later they did a practical training with the boards, in which they needed to solve exercises with and without help. In the end, the teachers were asked to answer a questionnaire about the workshop, the board and EasyLab. 83 teachers completed the questionnaire, of which a part of the statistical data is shown in Figure 9. 98.6 % of the teachers, who answered these questions, had the opinion, that the EasyLab programming interface is an advantage over the programming with C and Assembler. The same amount had the opinion that the integrated simulation tool is really helpful. Many of the teachers put additional information next to their answers, saying that with the board and especially with EasyLab it would be possible to concentrate on the task of teaching control theory and not programming. The teachers also recognized very quickly, that with the simulation tool, not every child or computer needs a board and that like this, schools can afford the system. Besides questions about the workshop and the EasyKit system, the teachers were also asked about their schools. An interesting result is the fact that 75 % of the teachers of the schools for professional education, but only 36.7 % of the teachers of schools for normal education, said, they would already use microcontrollers in their education. With 43.3 % the German school type "Gymnasium", which educates the future students, has a little bit of a higher percentage, but still is far away from schools for professional education. The reason given by some teachers is that in school there is no time to learn the required information about electronics and programming. But that means that students will have to learn all this information later besides their studies and when they have to learn it, they will have to start from the absolute basics. EasyKit, EasyLab and the didactical additional information can help to teach at least the basics of designing mechatronical systems.

3.4. **Replacing the starter board with the industrial version**

After getting to know the basics of microcontrollers, the user probably wants to advance. Because of its inflexible design, the starter board is restricted in the possible actions. With the knowledge about the programming of the starter board in EasyLab, the user can also program the industrial version of the electronics. It is more flexible and has more options. When knowing the starter board, it is actually not necessary to read the manual, but it is encouraged to do so. The datasheets of the electronics components are actually sufficient to put the system into operation, because they are well structured. The best option for getting used to the system is to start with a simple program, e.g. switching light emitting diodes with switches or analog inputs.

3.5. **Substituting the EasyKit electronics or the EasyLab for more flexibility**

As already mentioned, EasyKit can be used as development tool for mechatronical systems, but it is also a didactical system. Because of the didactical purpose, an option was left available, making it possible to replace the EasyKit system step by step. It is possible to start with any, either the electronics and continue to work with EasyLab or to replace EasyLab and continue to work with the EasyKit electronics. This might be necessary, if the limitations of the system, which result from the modular architecture, disturb the user.

When replacing any, either the electronics or EasyLab, additional programming hardware is required, because EasyLab is transferring the program to the microcontroller by a serial interface, which is included in the EasyKit system. The microcontroller contains a boot loader, which reads the new program from the serial interface and burns it to the memory. The programming hardware must be chosen according to the used microcontroller. When replacing the electronics components, an option in EasyLab has to be changed, the rest works like before.

4. SUMMARY

EasyKit is a good tool for getting into a first contact with microcontrollers. The additional didactical documentation, especially the web based training provides an easy to understand introduction, which contains interesting exercises, to keep the motivation of the user.

EasyLab leads very fast to a first success, when designing software. The main reason for this is the usage of graphical programming languages. Especially the combination of these languages in EasyLab and the starter board with its variety of basic functions, gives the user the ability to put the system into operation in a very short time.

The knowledge, learned during the usage of the EasyKit system, is important background information about microcontrollers, which can be very helpful, when designing mechatronical systems without EasyKit later.

5. ACKNOWLEDGEMENT

This work was funded by the German Ministry of Education and Research (BMBF) and coordinated by the Project Management Agency Forschungszentrum Karlsruhe (PTKA).

6. REFERENCES

[1] G. Bauer, R. Pittschellis, A. Knoll, „EasyKit – Entwicklungsmethodik für den Entwurf mechatronischer Systeme“, Mikrosystemtechnik-Kongress 2009, Berlin, Germany, Oct. 2009.

[2] R. Stetter, “Der Weg zum mechanischen Engeneering”, Technica, 19/2005, pp. 10-12.

[3] Arbeitsgemeinschaft Match-X, VDMA-Einheitsblatt VDMA 66305: „Bausteine und Schnittstellen der Mikrotechnik“, Verband Deutscher Maschinen- und Anlagenbau e.V., July 2005, Available: http://www.match-x.org.

[4] S. Barner, M. Geisinger, C. Buckl and A. Knoll, “EasyLab: Model-based development of software for mechatronic systems”, IEEE/ASME International Conference on Mechatronic and Embedded Systems and Application, Beijing, China, October 2008

[5] The MathWorks, “Simulink 7”, 2007, Available: http://www.mathworks.com/products/simulink/

[6] National Instruments, “LabVIEW 2009”, 2009, Available: http://www.ni.com/labview/

[7] P. Barnard, “Software Development Principles Applied to Graphical Model Development”, AIAA Modeling and Simulation Technologies Conference and Exhibit, San Francisco, USA, Aug. 2005

[8] Festo Didactic, “EasyKit macht Schule – Mechatronik einfach gemacht”, 2010, http://www.festo-didactic.com/de-de/lernsysteme/meclab-technik-fuer-allgemeinbildende-schulen/easykit-macht-schule-mechatronik-einfach-gemacht.htm

[9] STMicroelectronics, “STM32F – 32-bit ARM Cortex MCUs”, 2010, Available: http://www.st.com/mcu/inchtml-pages-stm32.html